"What is the future of testing? What would you advise testers to do to be prepared for that future? What should they learn?"

It's weird when a tester is asked to say something about the future.

The community is in an uproar. There are ongoing disputes about what we know, and about what we can know, about complex systems. There are serious questions about the trustworthiness of results from experimentation. Indeed, there are questions about whether experimentation is desirable at all. Some hold that the key to understanding our systems is to start with things that we know about, and put them together rigorously so that we can prove we understand what we're developing.

There is controversy over the role of tools that are expensive and unreliable. Those tools require extensive help from specialized consultants to maintain them. There are arguments and well-placed skepticism about whether those unreliable tools can be trusted to support our theories and practices. And, of course, because people are trying to figure out things that matter in a social context, there is intense friction based on personalities and styles of argumentation.

The situation I'm describing is the state of science in the 1660s, 350 years ago or so. The community was Britain's Royal Society and people in its orbit, including Robert Boyle, Robert Hooke, and Thomas Hobbes. The systems that they were trying to understand were natural systems, including properties of the air and of gasses. The product that they were building was knowledge about the natural world. One of the expensive and unreliable tools was the air pump, built by Hooke to support Boyle's experimental approach. One of the skeptics was Hobbes, who raised serious objections about the trustworthiness of the tools, the validity of the experiments, and the conclusions that people could draw from them. Boyle and his colleagues were compelled to answer those objections.

One product of that controversy—and others over the next 350 years or so, through to the present day—is the scientific method itself, by which we continually revise our knowledge of the world and the things in it. Science evolved. It took almost 300 years for scientists and philosophers to agree that experimental results could provide support for a theory, but could not prove that a theory was correct.

Another product of the controversy is our current state of the art in what we know about the world. That knowledge, whether we like it or not, is always provisional—what we know so far, based on models we develop, with help from the tools we have. There are still uncertainties and arguments over the state of nature, and what we can say about it.

Testing today seems to me to be a lot like science in the early days. We struggle not only to understand things, but also to decide on how we come to understand things. Some testers believe that testing is about making sure that the product works "as expected", confirming what we know or what we would like to believe. This, I think, leads to pretty shallow testing that whistles past the graveyard of risk.

Some of us—relatively few, alas, so far—believe that what we anticipate about the product is mostly beside the point; what we discover about it is paramount. We must explore the product in a way that targets and confronts risk. To do so, we must not only interact with the product in routine ways, but we must also set up experiments that, by design, explore outside of what we think of as everyday experience.

In addition to observing the running product directly, we must also investigate the details and the surrounding systems, just as the scientists of the 17th century began to look at things that were very small or far away, and how they related to each other. In addition to normal use, we must subject the product to extreme conditions, overwhelming it or starving it or getting rid of something essential, just as the early scientists did with air. In addition to observing things in the lab, we must go into the field as Darwin did. We must not only question our products, but we must also question how testing happens, just as Kuhn and his followers questioned the proceedings of science.

The idea behind all this is not to confirm, but to challenge what we believe. We are not here to show that the product is fine, but to discover

problems that threaten the value of the product and the project, at every level of granularity. We do this to help our clients to decide whether product we've got is the product we want, rather than the product we hope we've got.

So: to prepare for the future, one thing that great testers could do is to study the past—the history of human successes and failures. Testers must be generalists and polymaths, aware of elements of science, technology, economics, psychology, mathematics, anthropology, engineering, sociology, and the humanities—art and design and literature, and music too. Every tester must also be to some degree a specialist, focusing on things that are interesting and valuable to them and to their projects. As Karl Weick said, "if you want to understand something complicated, you have to complicate yourself." The one prediction I think I can make about the future is this: things will become more complicated.

"What is the future of testing, which role is there for testers? In other words: what will be their daily practice?"

One key difference between the physical sciences and testing is that the complex systems are being produced not by Mother Nature, but by us. Unlike Nature, people are impatient and driven by ambition. As a result, we tend to have a short-term focus, often building and trying to sustain systems that are not viable. Nature takes the long view, and dispassionately arranges for the weak ones to be killed early. Nature isn't worried about getting in trouble from the senior management team.

Economics, ambition and invention will drive the creation of new products and services. People are inventive, clever, and diligent. Yet there will always be risk that bad things will happen, and that good things won't happen, because people are also to some degree naive, incompetent, and careless. As long as that risk is consequential, there will be a need for critical thinking, systems thinking, risk analysis, experimentation; for discovery, investigation, and learning; and for rapid reporting and feedback into the development process.

Everyone involved in developing a product or service should take on those tasks from time to time. However, the effort to examine the product under a wide variety of conditions can swamp the builder's building time, and delay useful feedback. Moreover, excellent testing requires critical distance—the difference between one perspective and another—and the switch from the builder's mindset to the tester's mindset can be particularly difficult or time-consuming. One way around that is to have people dedicated to testing and the skills that it requires. The daily task for skilled testers will be to challenge the plans and designs for the product, and then to explore and experiment with the built product to find the challenges and risks and problems that have so far been overlooked.

Tools and trappings and daily practices around testing will change—and that's a good thing—but the basic theme will not. I hope that more people will come to recognize that testing thinking and practice can be applied at any stage, and that "product" refers to anything that someone has produced. Review is a form of testing in which we explore ideas and artifacts, and perform thought experiments on them. That said, ideas and intentions and designs and representations are one thing; built code and running systems another, so we must investigate them and perform experiments on them too. Testing must be focused on the idea that whatever we think we've built, the real thing can fool us. □

Michael Bolton is a consultant, software tester, and testing teacher who helps people to solve testing problems that they didn't realize they could solve. In 2006, he became co-author (with James Bach) of Rapid Software Testing (RST), a methodology and mindset for testing software expertly and credibly in uncertain conditions and under extreme time pressure. Since then, he has flown over a million miles to teach RST in 35 countries on six continents.
Michael has 25 years of experience testing, developing, managing, and writing about software. For the last 18 years, he has led DevelopSense, a Toronto-based testing and development consultancy. Prior to that, he was with Quarterdeck Corporation for eight years, during which he managed the company's flagship products and directed project and testing teams both in-house and around the world.

Contact Michael at **michael@developsense.com**, on Twitter **@michaelbolton**, or through his Web site, **http://www.developsense.com**.